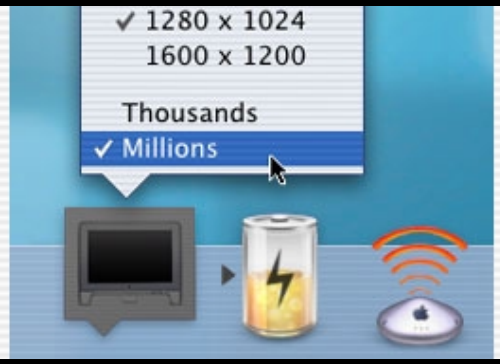
Session 702

Interface Builder



Henri Lamiraux
Engineering Manager

Interface Builder

- Wide audience with different interests
- Talk organized around
  - Frequently asked questions
  - New features
  - Tips
  - Demos

Roadmap

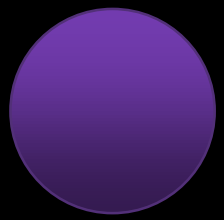
- What is Interface Builder (IB for short)
- Developing an application with Interface Builder
  - Laying out the user interface
  - Connecting the user interface to code
  - Loading the user interface
- Tips and Tricks
- nibtool
- Update Schedule

Roadmap

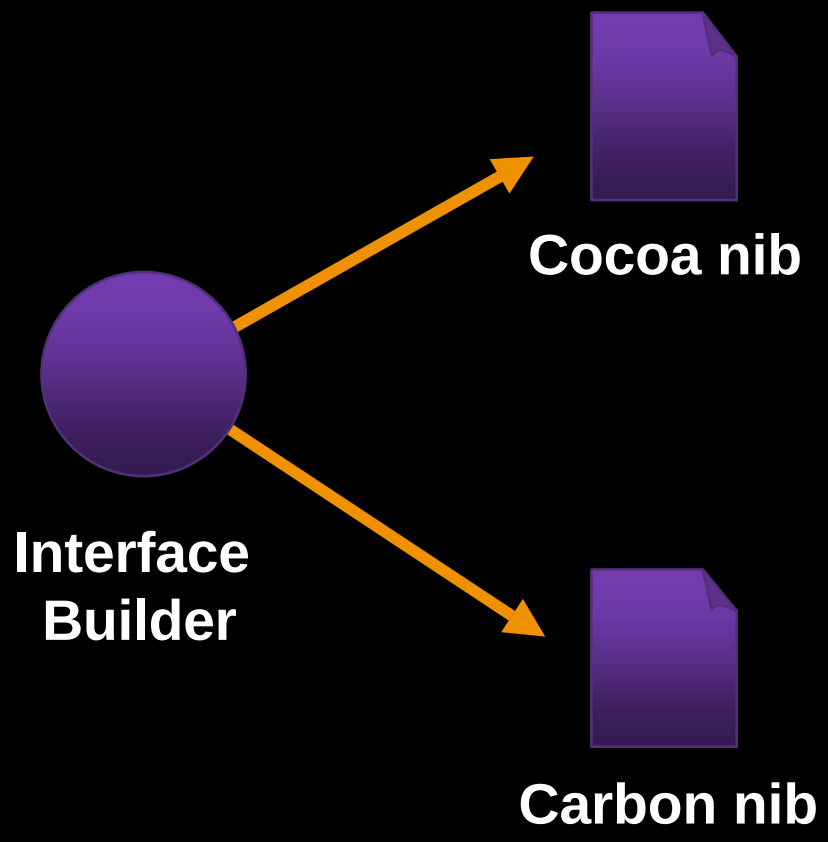
- What is Interface Builder
- Developing an application with Interface Builder
  - Laying out the user interface
  - Connecting the user interface to code
  - Loading the user interface
- Tips and Tricks
- nibtool
- Update Schedule

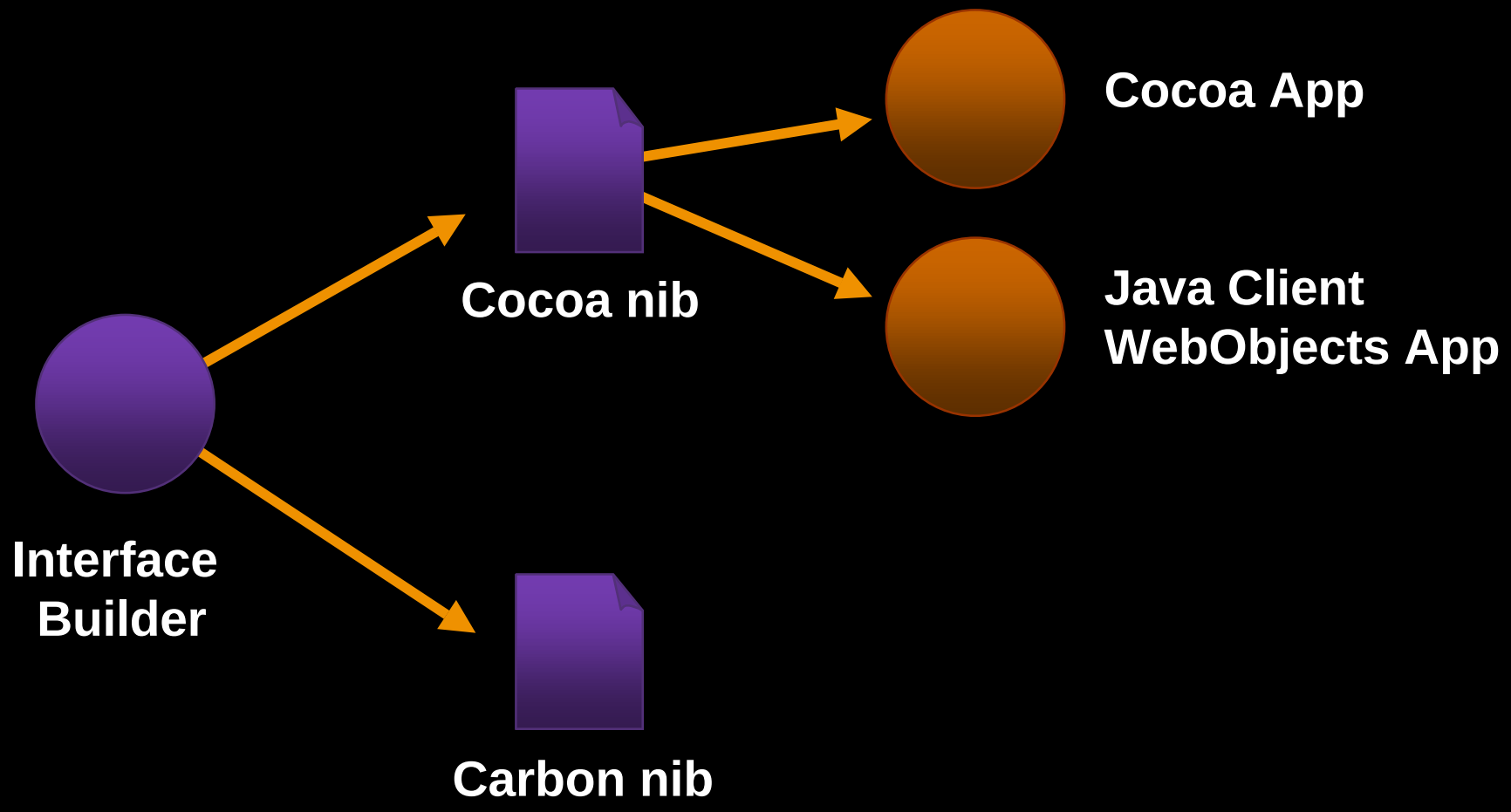
What Is Interface Builder

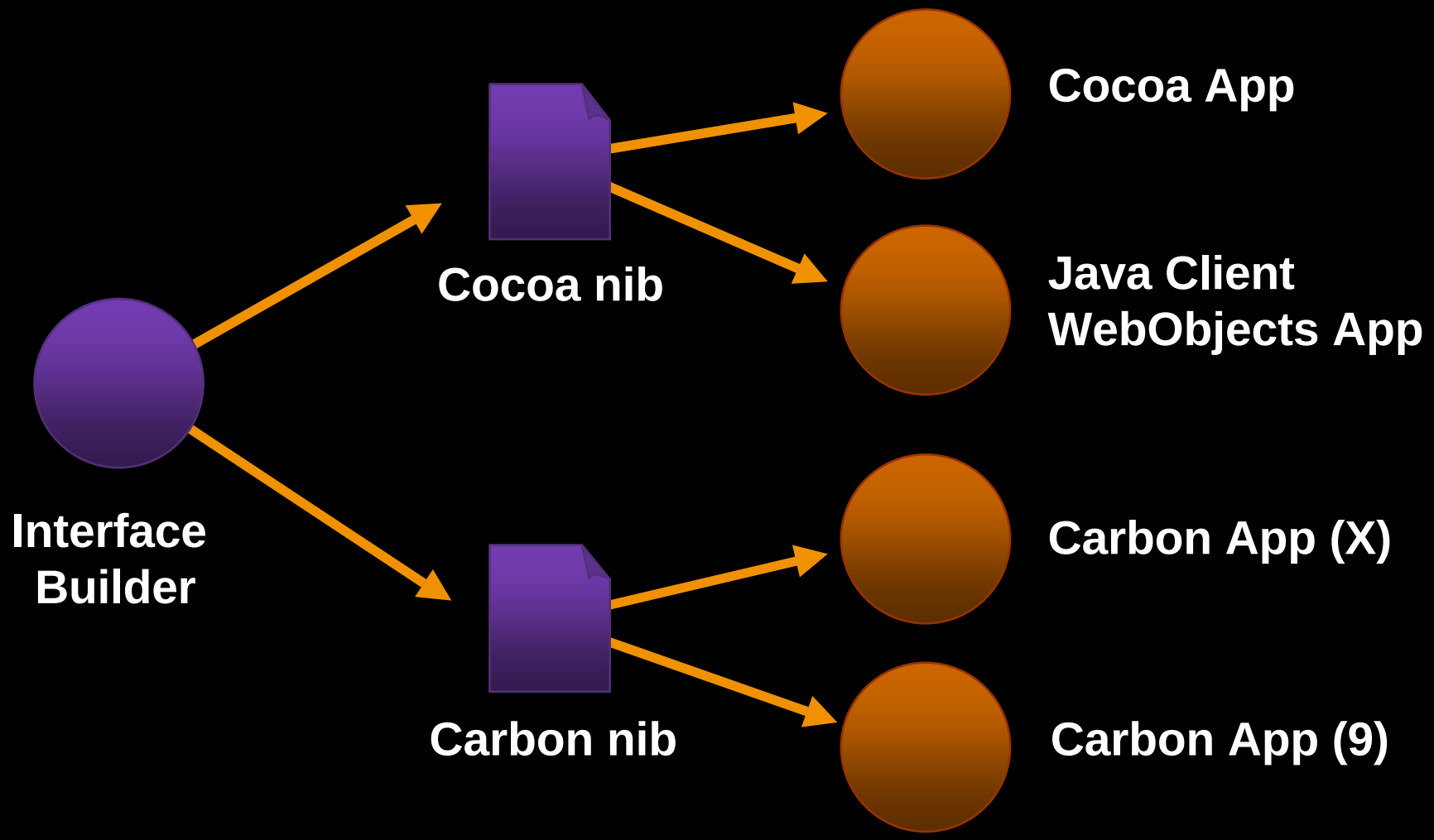
- Apple's User Interface Builder Tool
- Create Aqua compliant User Interfaces for Cocoa and Carbon



Interface Builder


Cocoa nib
Interface Builder
Carbon nib


Cocoa App
Cocoa nib
Java Client
WebObjects App
Interface
Builder
Carbon nib


Cocoa App
Cocoa nib
Java Client
WebObjects App
Interface
Builder
Carbon App (X)
Carbon nib
Carbon App (9)

Roadmap

- What is Interface Builder
- Developing an application with Interface Builder
  - Laying out the user interface
  - Connecting the user interface to code
  - Loading the user interface
- Tips and Tricks
- nibtool
- Update Schedule

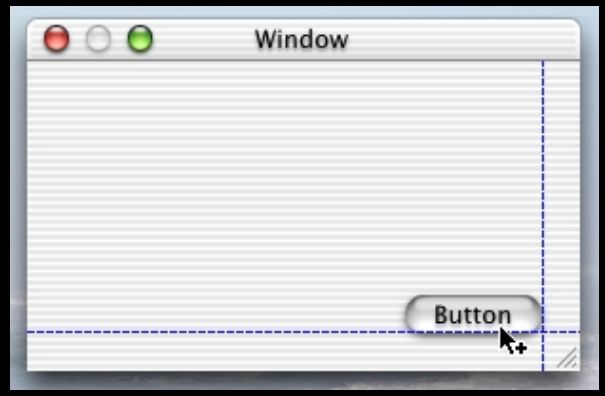
Laying Out the User Interface

- Guides
  - Replacement for the Grid
  - Much faster and intuitive way to align objects
  - Two types:
    - Aqua Guides
    - User Guides

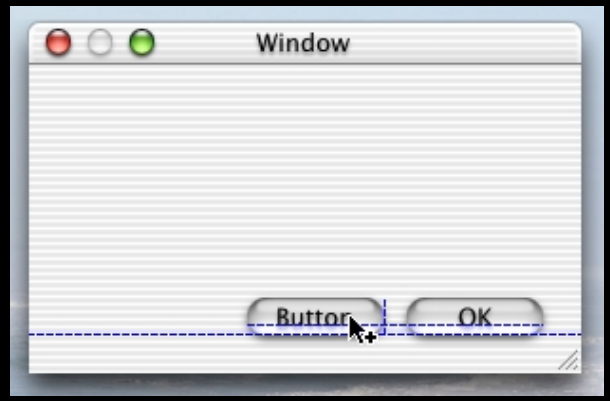
Laying Out the User Interface

- Aqua Guides
  - Implements Aqua layout guidelines

(1)



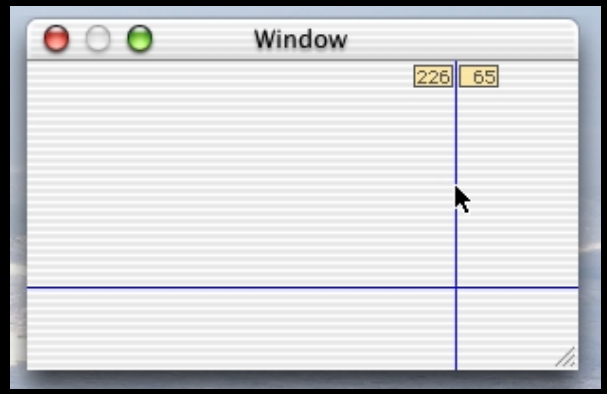
(2)



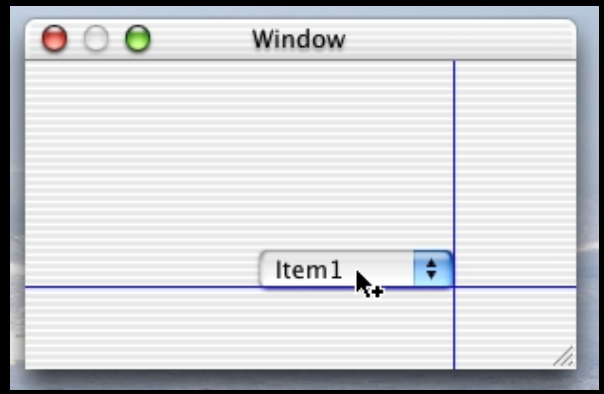
Laying Out the User Interface

- User Guides
  - User defined guides
  - Saved with the nib file

(1)

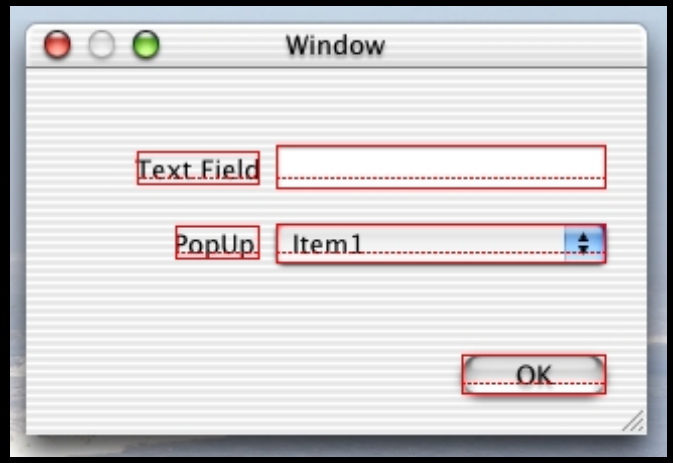


(2)



Laying Out the User Interface

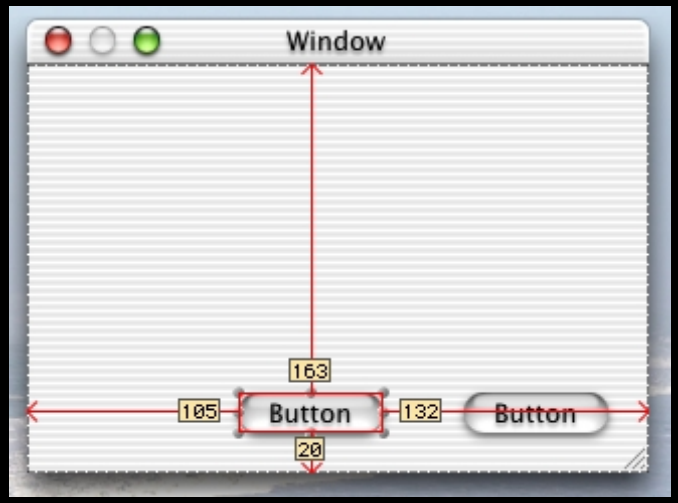
- Layout Rect
  - Visual bounding box of an object (without shadow)
  - Matches Aqua Guidelines
  - Show/Hide with Command-L



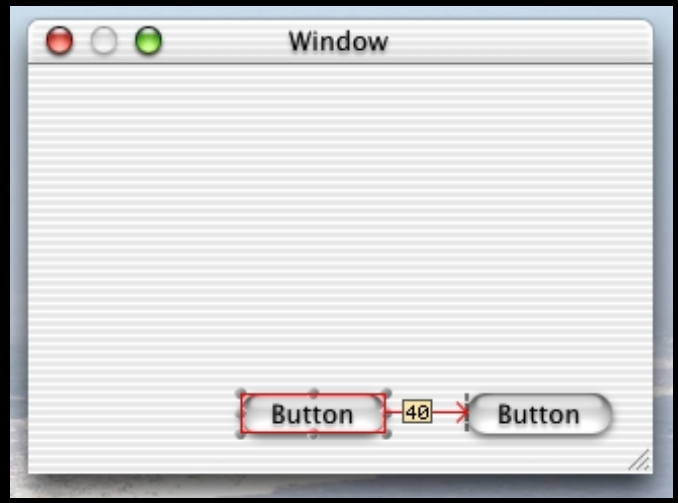
Laying Out the User Interface

- Distance Measurement
  - Show distance between selected object and object under the cursor
  - Triggered with Option Key

(1)

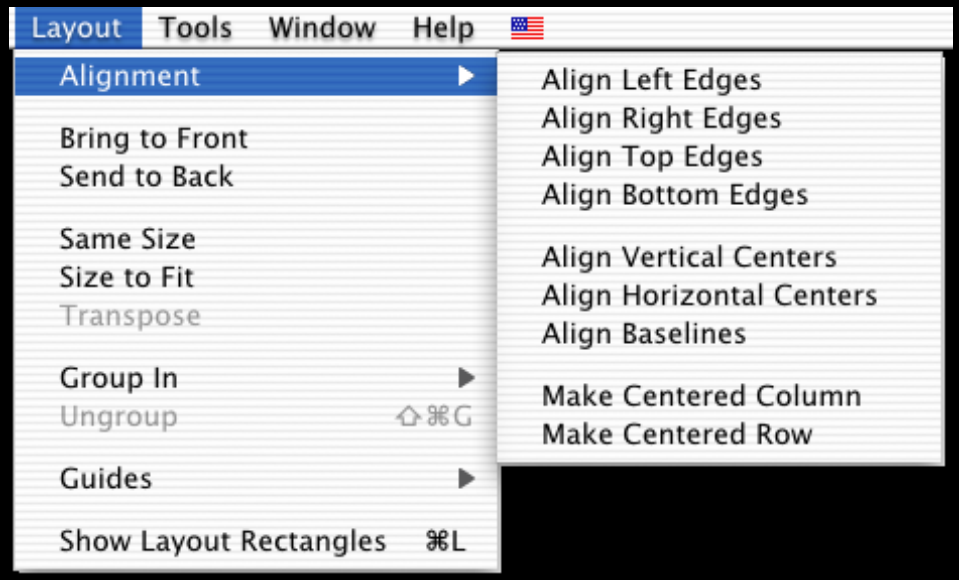


(2)



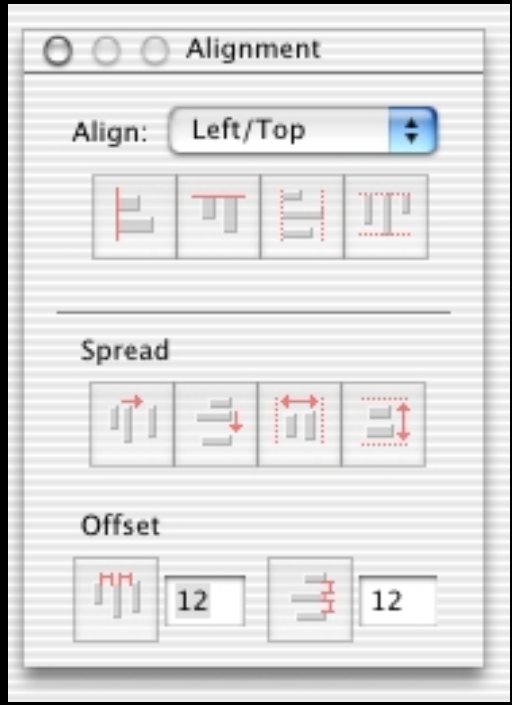
Laying Out the User Interface

- Layout Menu
  - Act on group of objects



Laying Out the User Interface

- Alignment Panel

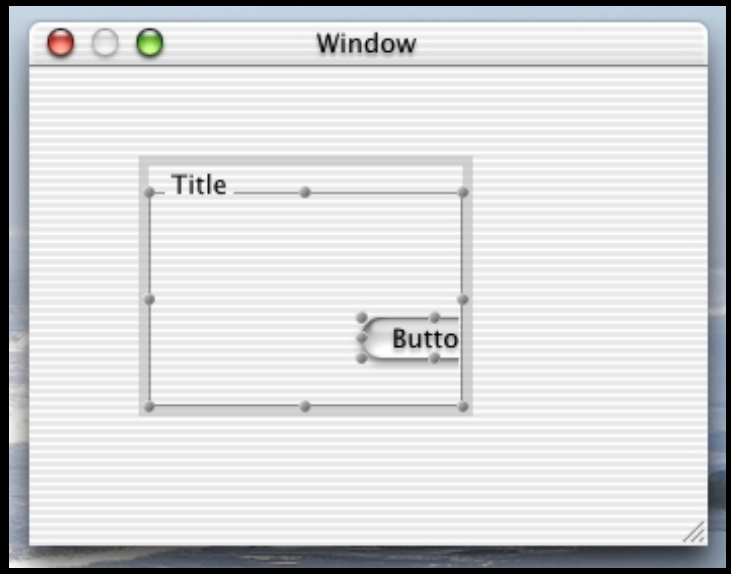


Laying Out the User Interface

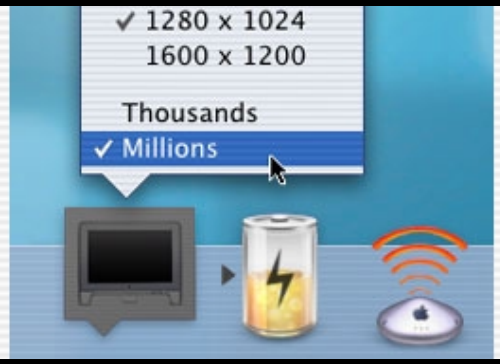
- Test Mode
  - Simulate a running application within IB
  - For both Cocoa and Carbon

Laying Out the User Interface

- Containers
  - IB maintains a hierarchy of UI elements



Demo



Scott Herz
Interface Builder Engineer

Roadmap

- What is Interface Builder
- Developing an application with Interface Builder
  - Laying out the user interface
  - Connecting the user interface to code
  - Loading the user interface
- Tips and Tricks
- nibtool
- Update Schedule

Connecting the UI to Code

- Different programming models
  - Carbon: Procedural (C APIs)
  - Cocoa: Object Oriented
    - Relies on dynamic features of Objective-C

Connecting UI to Code—Carbon

- IB doesn't change the programming model
  - Better way to layout and package the user interface
  - Easy to take advantage of the Carbon Event Model

Connecting UI to Code—Carbon

- Getting a reference to UI elements
- Reacting to user actions

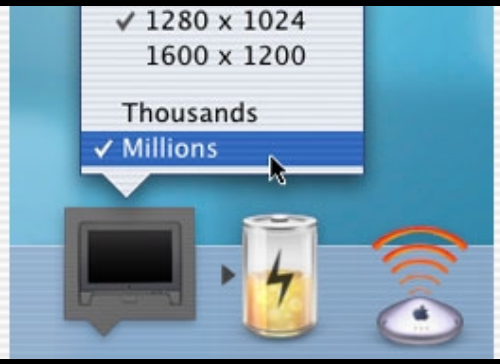
Connecting UI to Code—Carbon

- Getting a reference to UI Elements
  - In Interface Builder assign:
    - Control ID to controls
    - Menu ID to menus
  - In your code, call the appropriate API
    - GetControlByID(…), GetMenuRef(…), etc…

Connecting UI to Code—Carbon

- Reacting to user actions
  - In Interface Builder assign a Command ID to
    - Menu Items
    - Controls
  - Implement Event Handlers to process those commands
    - Attend session 106 “Carbon Event Manager”
      - Today at 5:00pm in Hall 2

Demo



Scott Herz
Interface Builder Engineer

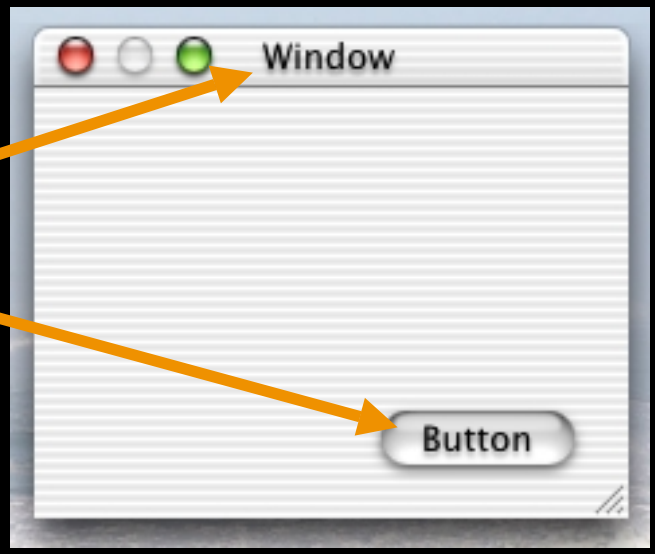
Connecting UI to Code—Cocoa

- Four important Cocoa/IB concepts:
  - Outlet
  - Target/Action
  - File's Owner
  - First Responder

Connecting UI to Code—Cocoa

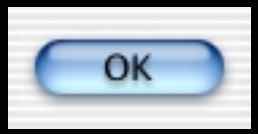
- Outlet
  - An instance variable that points to another object

```
@interface foo {
        NSWindow* window;
        NSButton* button;
}
....
@end
```



Connecting UI to Code—Cocoa

- Target
  - An object responsible for responding to the user's action
- Action
  - The method that specifies what the target is to do



Target

Action

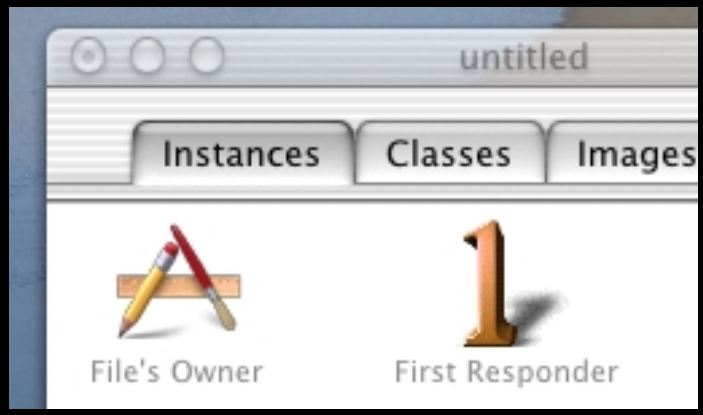
```
@implementation foo
...
- (IBAction)doSomething:(id)sender
{
....
}
...
@end
```

Connecting UI to Code—Cocoa

- Interface Builder's Role
  - Graphically connect Outlets and Targets/Actions
  - Connections are saved in the nib file
  - At runtime, everything is hooked back up
    - Outlets are set
    - Controls/menu items are ready to trigger the appropriate code

Connecting UI to Code—Cocoa

- File's Owner and First Responder
  - Proxy objects: stand for some other object



Connecting UI to Code—Cocoa

- File's Owner
  - Existing object, external to the nib
  - Specified when loading the nib

```
[NSBundle loadNibNamed:@"myNib"owner:theOwner];
```

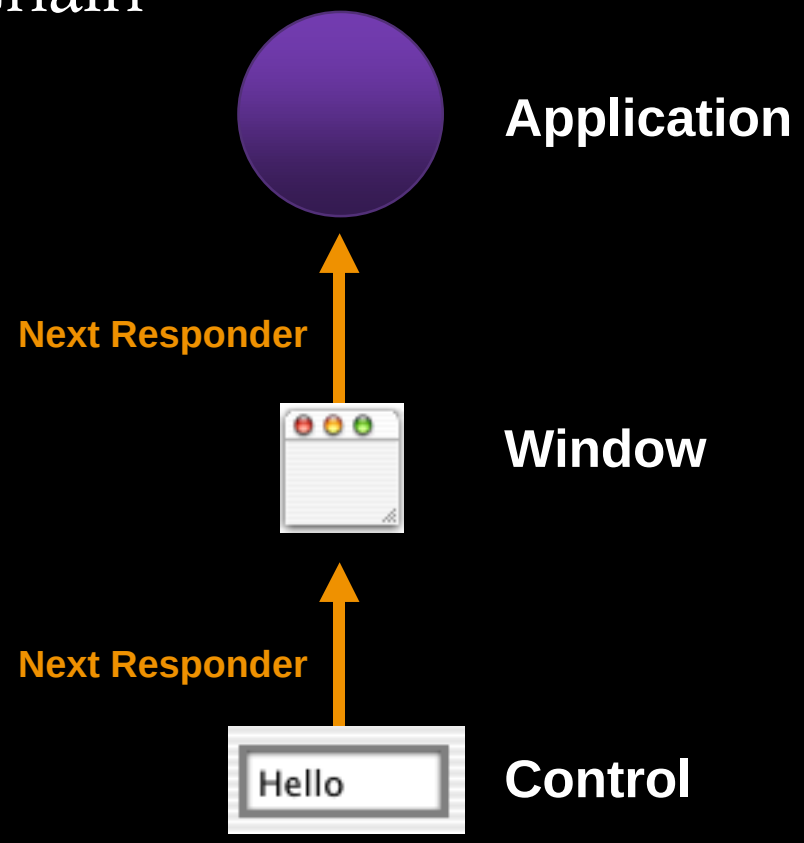
  - Channels messages between objects unarchived from the nib file and other objects in the application

Connecting UI to Code—Cocoa

- First Responder
  - More of a status conferred to an object than a real object
  - First object in the responder chain to respond to a message

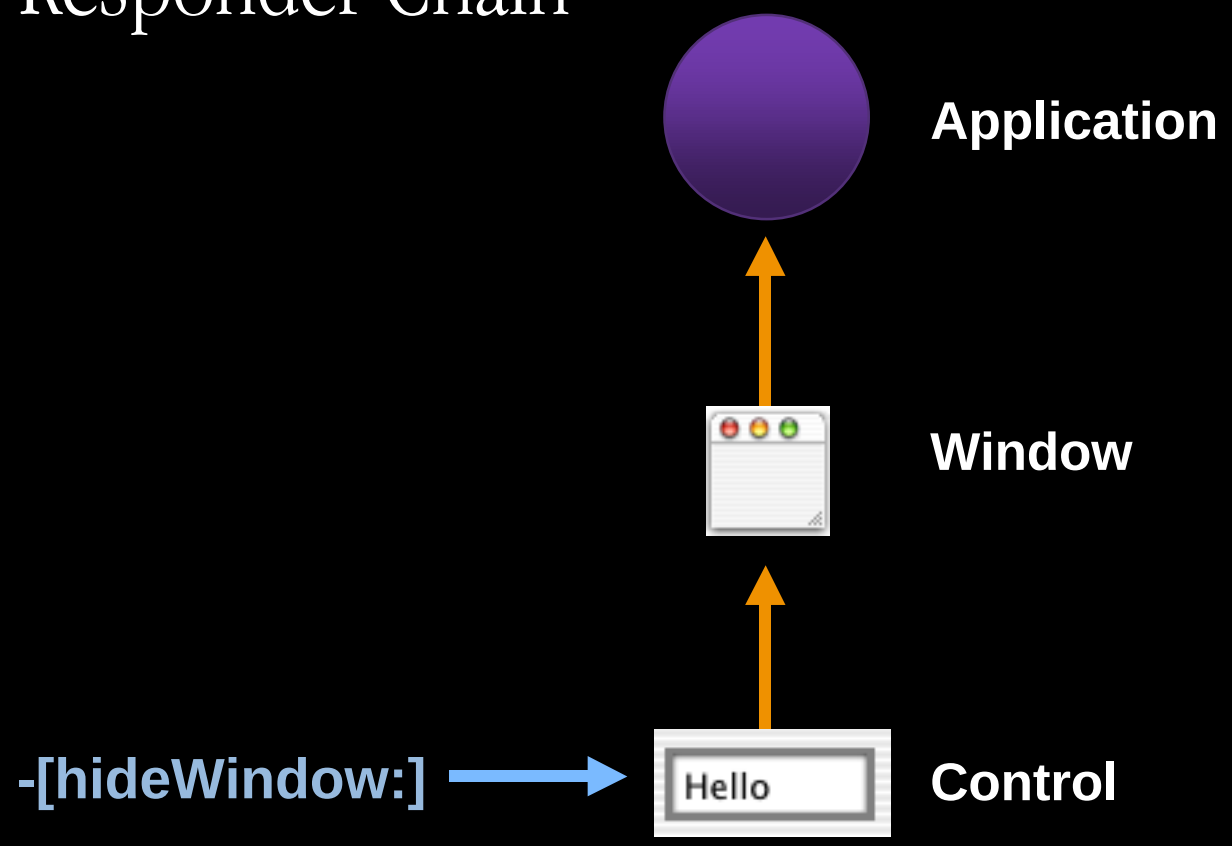
Connecting UI to Code—Cocoa

- Responder Chain



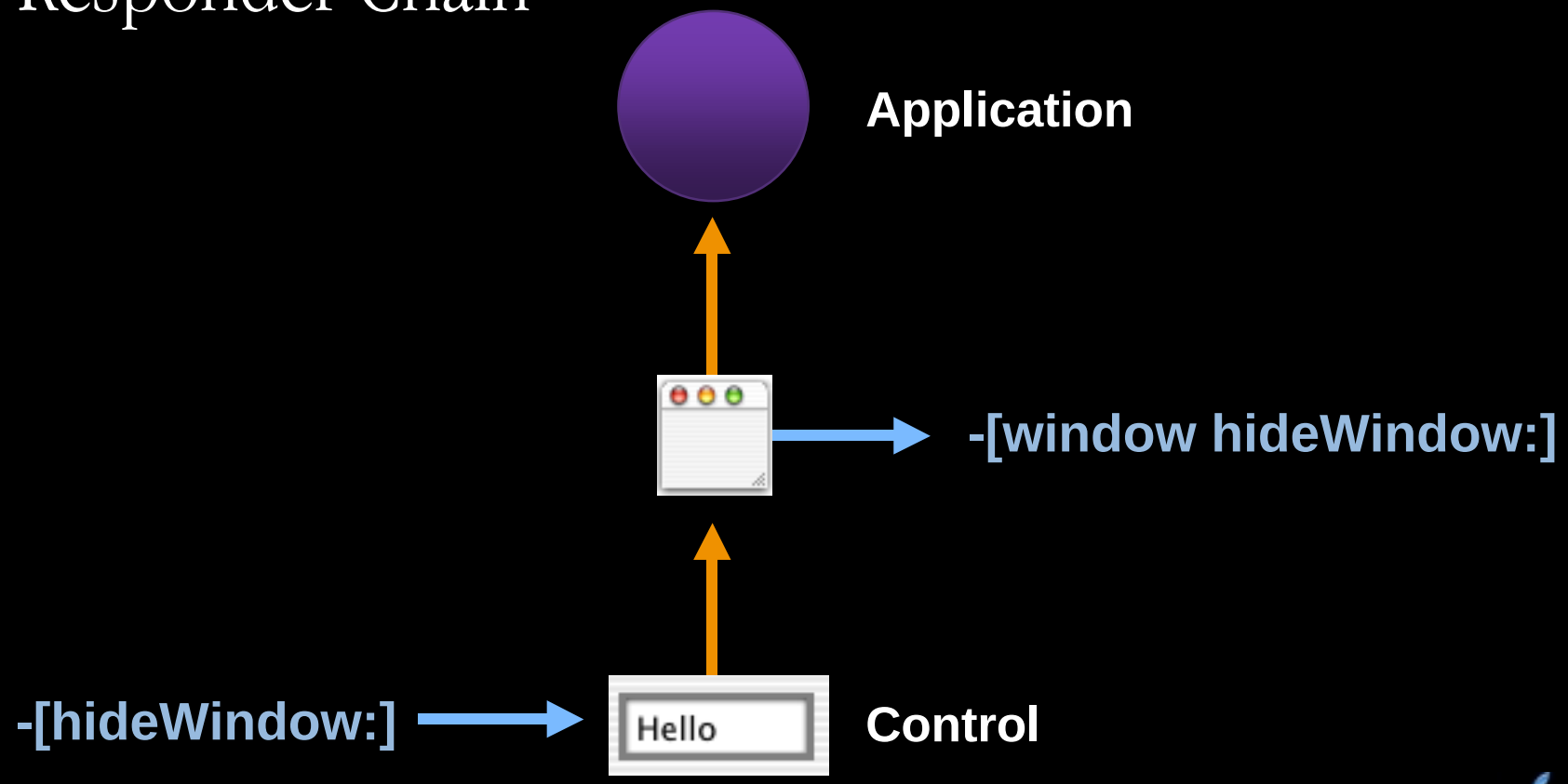
Connecting UI to Code—Cocoa

- Responder Chain



Connecting UI to Code—Cocoa

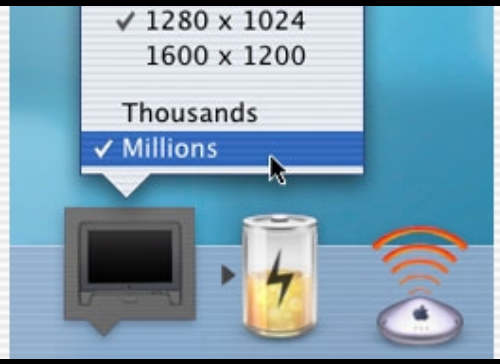
- Responder Chain



Connecting UI to Code—Cocoa

- First Responder is useful to connect an action:
  - To an unspecified instance of a target
    - Cut/Copy/Paste
    - Hide Window
    - Etc…
  - To an object in another nib
    - As long as this target object is in the responder chain

Demo



Scott Herz
Interface Builder Engineer

Roadmap

- What is Interface Builder
- Developing an application with Interface Builder
  - Laying out the user interface
  - Connecting the user interface to code
  - Loading the user interface
- Tips and Tricks
- nibtool
- Update Schedule

Loading the User Interface

- Different approach between Cocoa and Carbon
  - Carbon
    - Always explicit
  - Cocoa
    - Automatic for the “main” nib
    - More or less automatic for other nibs

Loading the UI—Carbon

- One Header: IBCarbonRuntime.h
- 6 Functions
- On Mac OS X: part of Carbon framework
- On Mac OS 9: part of CarbonLib (v 1.1 or later)

IB Carbon Runtime APIs

```
int main(int argc, char* argv[]) {
    IBNibRef nibRef;
    WindowRef window;

    CreateNibReference (CFSTR("Main"), &nibRef);

    SetMenuBarFromNib (nibRef, CFSTR("MenuBar"));

    CreateWindowFromNib (nibRef,
            CFSTR("MainWindow"), &window);

    DisposeNibReference (nibRef);

    ShowWindow(window);
    RunApplicationEventLoop();
}
```

Loading the UI—Cocoa

- One special nib file: the “main” nib
  - Automatically loaded at launch time
  - Name of Main nib set in Project Builder (“Application Settings” tab)
  - Normally contains the menu bar
  - File’s Owner will be the application object

Loading the UI—Cocoa

- Other nibs should be loaded lazily
  - Explicitly using
    - -[NSBundle loadNibNamed:owner:]
  - Using NSWindowController
    - Whenever possible
    - File's Owner will be the windowController
    - Check new "SimpleMultiWindow" example in /Developer/Examples/Interface Builder

Roadmap

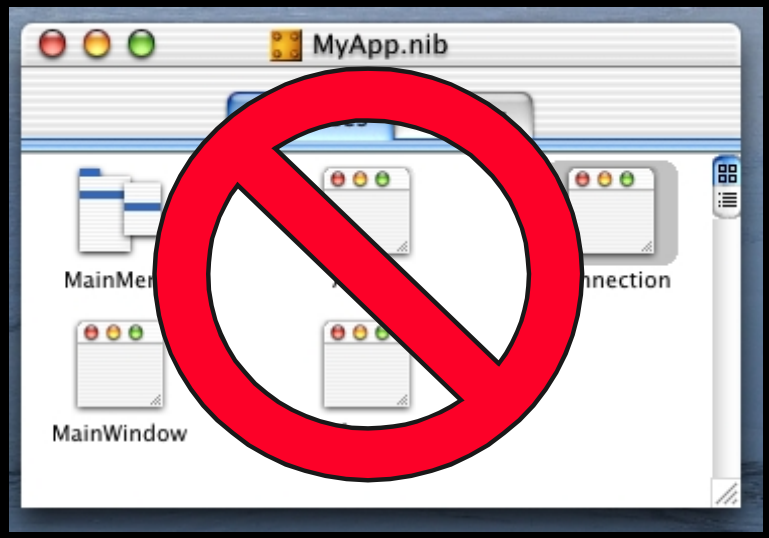
- What is Interface Builder
- Developing an application with Interface Builder
  - Laying out the user interface
  - Connecting the user interface to code
  - Loading the user interface
- Tips and Tricks
- nibtool
- Update Schedule

Tips and Tricks

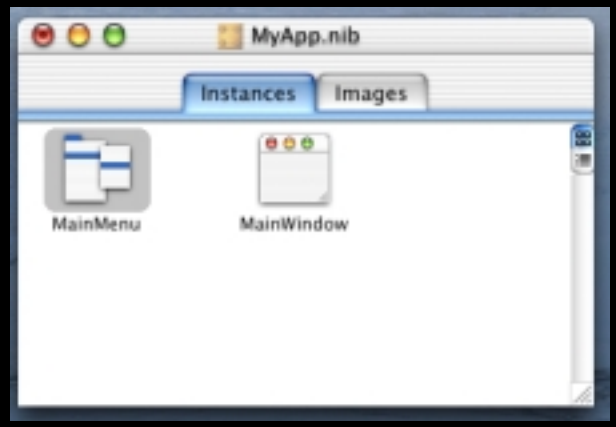
- Don’t put your entire UI in one nib
  - All objects in a nib file are instantiated at once
  - Your nib files will load faster if broken up
  - Facilitates reuse

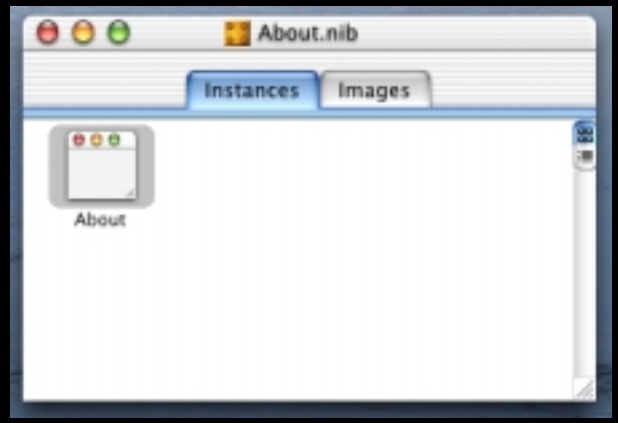
Tips and Tricks

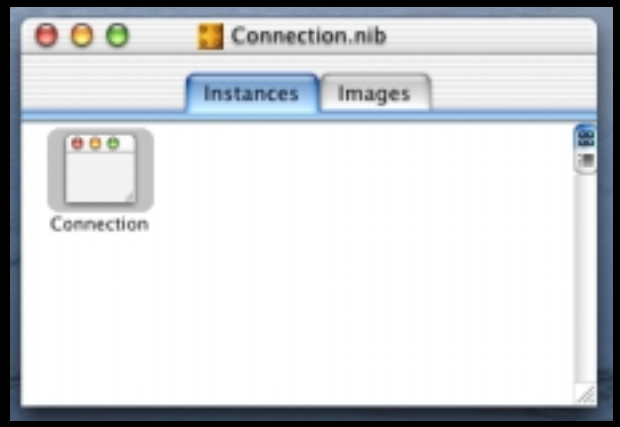
Tips and Tricks

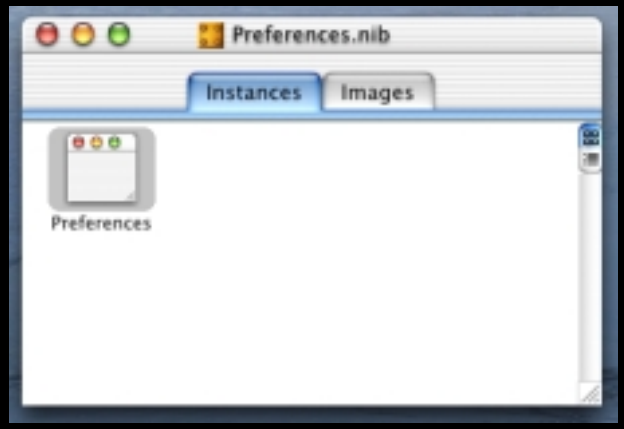


Tips and Tricks



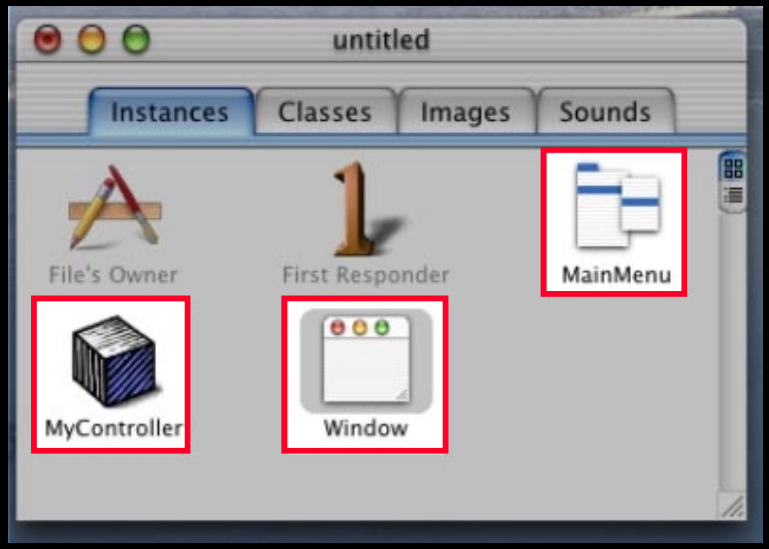






Tips and Tricks—Cocoa

- "Top Level Objects"
  - Windows
  - MainMenu
  - Controllers



Tips and Tricks—Cocoa

- Additional Top Level Objects
  - Formatters
  - Views (Windowless Views)
  - Single Menu (for Contextual Menus)
    - Also available for Carbon

Tips and Tricks—Cocoa

- Top Level Objects Reference Count
  - Reference count of one
  - Need to be released by some other object (usually the File's Owner)
    - NSWindowController does it for you
  - One exception:
    NSWindow with the "release on close" attribute

Tips and Tricks—Cocoa

- Windowless view
  - Allows a view to be instantiated without an enclosing window
    - Drag a Custom View to the document window
  - Eliminates the need to create a window and use only the content view
  - Used for the content view of NSDrawer
  - Be careful with reference counting
  - First implementation in Cocoa. Carbon in a future release

Tips and Tricks—Cocoa

- Post nib loading initialization
  - Implements -[awakeFromNib]
    - All objects have been instantiated
    - All outlets, targets/actions have been set
  - Don't rely on -[init], -[initWithFrame:], or -[initWithCoder:]

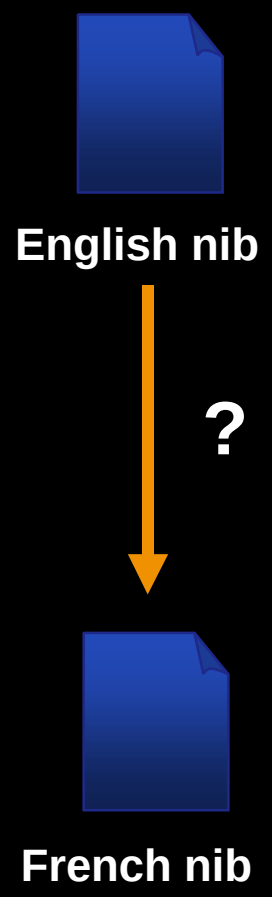
Roadmap

- What is Interface Builder
- Developing an application with Interface Builder
  - Laying out the user interface
  - Connecting the user interface to code
  - Loading the user interface
- Tips and Tricks
- nibtool
- Update Schedule

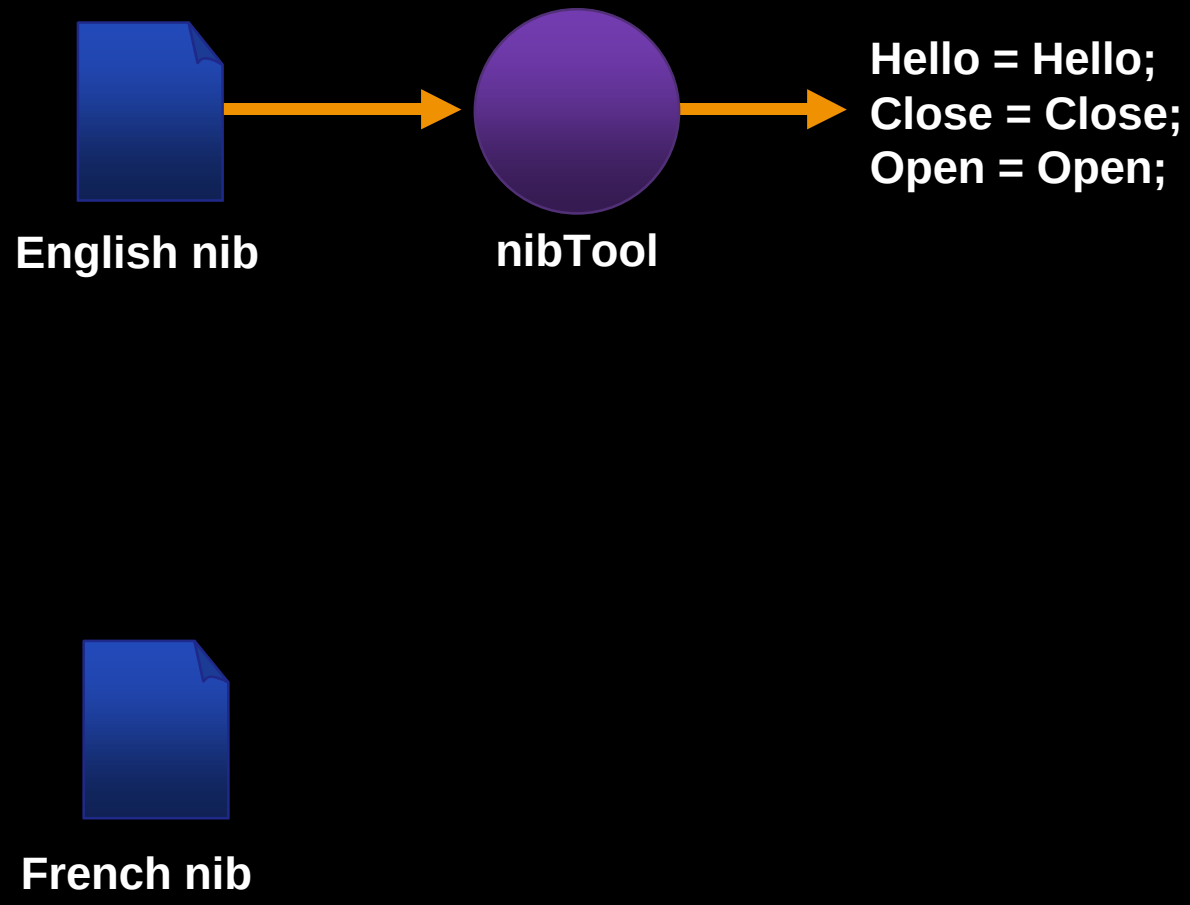
nibtool

- Lets you manipulate the contents of a nib file from the command line
- Works with both Cocoa and Carbon nibs
- Look at nibtool manpage for all options
- Used by File Merge to diff nib files
- New: nib localization support

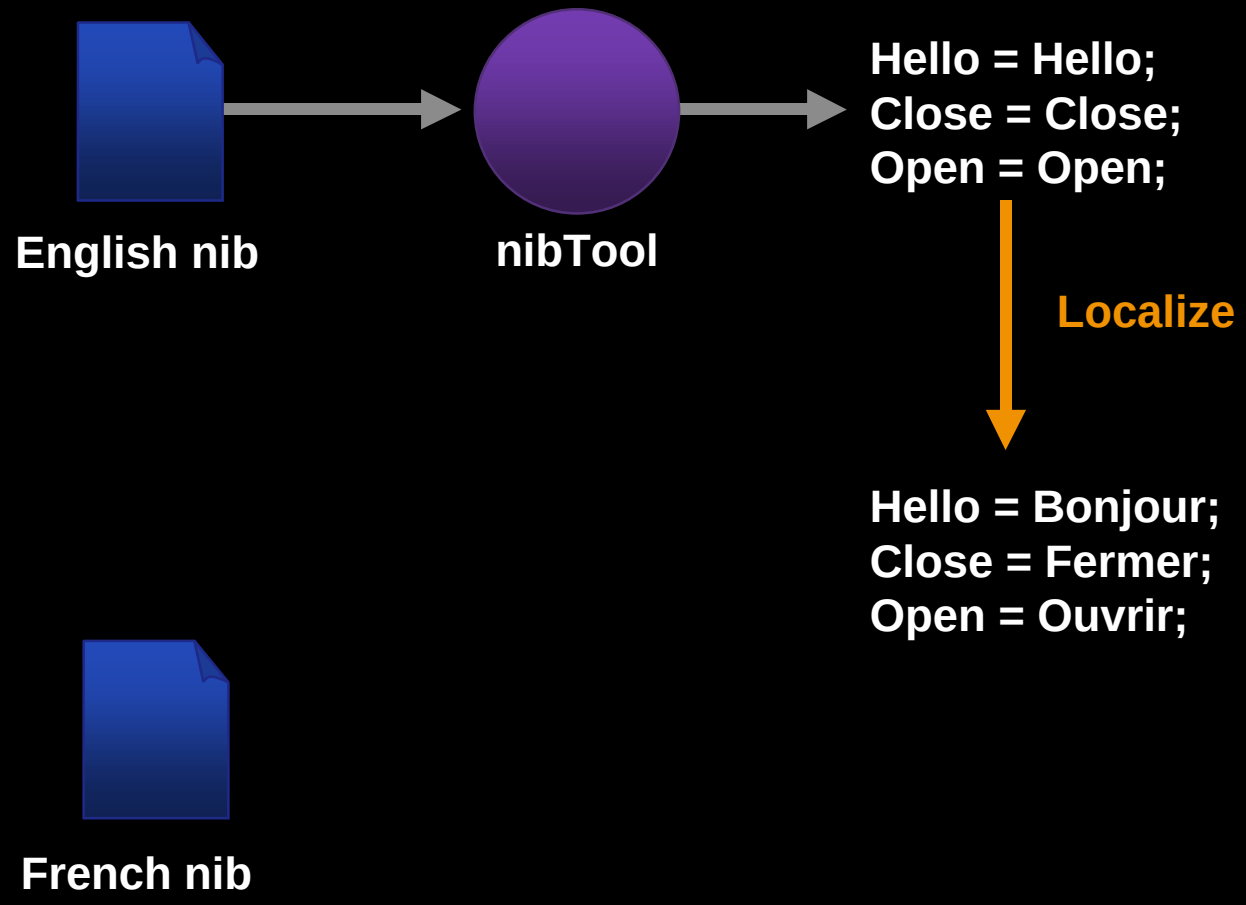
Localization With nibtool



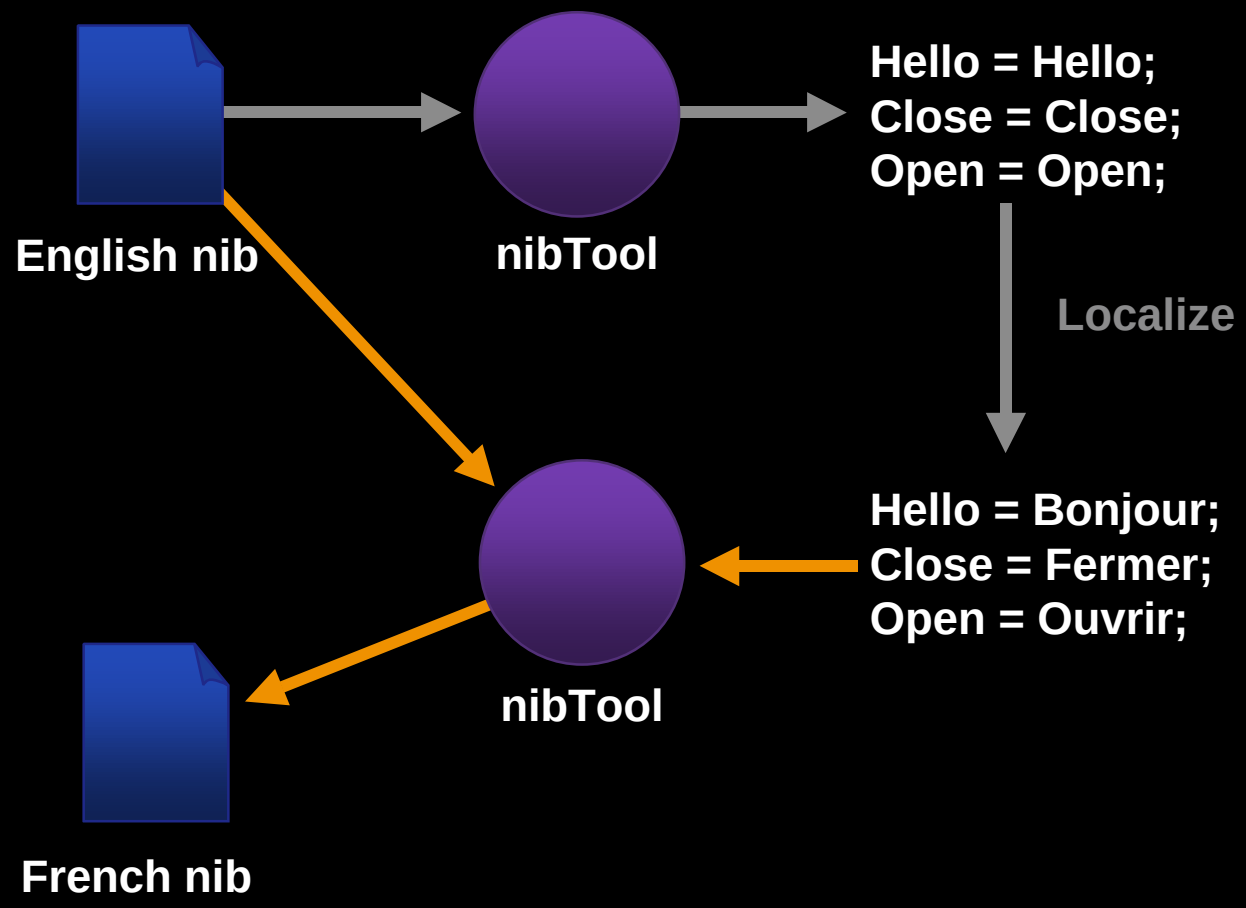
Localization With nibtool



Localization With nibtool



Localization With nibtool



Localizing Your Application

- Attend:

129 Mac OS X Application Localization	Room C **Thurs., 3:30pm**

Roadmap

- What is Interface Builder
- Developing an application with Interface Builder
  - Laying out the user interface
  - Connecting the user interface to code
  - Loading the user interface
- Tips and Tricks
- nibtool
- Update Schedule

Update Schedule

- Mac OS X Developer CD: IB version 2.0
- Version 2.0.1 is shipping at Conference
  - Stability
  - Bug fixes
  - Two new examples
    - BusyPalette
    - SimpleMultiWindow
- Quarterly Developer update

Update Schedule

- Future releases
  - Continue to concentrate on stability
  - Better view editing, more intuitive editing of containers
  - New UI for defining outlets and actions (Cocoa)

Update Schedule

- Future releases (Cont.)
  - New Class tab (Cocoa)
  - Flagging of unconnected Outlets and Targets/Actions (Cocoa)
  - Improved PB/IB integration
  - Closely track Cocoa and Carbon new features
  - And more…

Information Resources

- Information pages for all Mac OS X tools
  http://developer.apple.com/tools/
- Project Builder web page
  http://developer.apple.com/tools/projectbuilder/

- Apple's mailing list master page
  http://lists.apple.com/

  Mailing lists:
  - **projectbuilder-users**
  - **cocoa-dev**

Roadmap

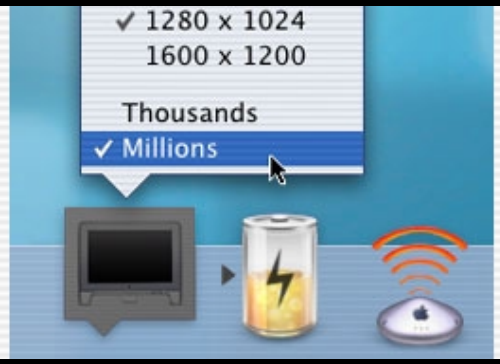
106 Carbon Event Manager	Hall 2 **Tues., 5:00pm**
122 Using Cocoa	Hall 2 **Thurs., 3:30pm**
123 Advance Cocoa Topics	Hall 2 **Thurs., 5:00pm**
129 Mac OS X Application Localization	Room C **Thurs., 3:30pm**

Who to Contact

Godfrey DiGiorgi
Technology Manager, Development Tools
Apple Worldwide Developer Relations
ramarren@apple.com

Macosx-tools-feedback@group.apple.com
Mailing address for feedback to Apple

Q&A



Godfrey DiGiorgi
Technology Manager for Development Tools
ramarren@apple.com

http://developer.apple.com/events/urls.html

WWDC

Worldwide Developers Conference 2001